

PostScript und PDF

PostScript – Chance oder Chaos?

„Ach bitte, könnten Sie mir mal schnell die PostScript-Datei ausdrucken?“ – „Ja, natürlich! Wie und unter welchem Betriebssystem ist sie denn erstellt?“ – „Das weiß ich nicht, ist schon länger her, wurde aber schon mal gedruckt. Ist ja PostScript, muß doch gehen! Es soll ein A0 Poster werden, in Farbe natürlich!“

So oder ähnlich werden die Kolleginnen unserer Abteilung sehr häufig gebeten, das Unmögliche möglich zu machen.

Allgemeines

PostScript (PS) wurde 1983 von der Firma Adobe Systems als Seitenbeschreibungssprache entwickelt. Hinter PostScript verbirgt sich eine Programmiersprache mit Grafikfunktionen, die in der Lage ist, geometrische Objekte, Schriften und Rasterbilder in einem Dokument anzuordnen. Die wichtigste Eigenschaft einer solchen PS-Datei ist die Bereitstellung einer normierten, geräteunabhängigen Schnittstelle für die Ausgabe solcher Dateien auf Drucker, Kopierer, Bildschirm, Belichtungs- oder Publishinggeräten. Solche Ausgabegeräte arbeiten meist rasterorientiert. Daraus folgt, daß die erzeugte Seite in Bildpunkte umgewandelt werden muß. Dazu benötigt man einen PS-Interpreter, auch RIP (Raster Image Processor) genannt, der die PS-Anweisungen analysiert und sie dann in die Sprache umwandelt, die das Ausgabegerät versteht.

Erzeugung von PostScript-Dateien

Es gibt viele Möglichkeiten, PS-Dateien zu erzeugen. Vier der wichtigsten sollen kurz beschrieben werden. Als erstes kann man die Sprache PostScript erlernen und seinen PS-Quelltext erzeugen, der mit lesbaren Anweisungen den Aufbau einer auszugebenden Seite beschreibt. Der Quelltext besteht aus einem Prolog, der Strukturkommentare, Autor, Titel, Seitenanzahl, PS-Version usw. enthält, und einem Skript [Abb.1], mit dem die Seite beschrieben wird.

Als zweites kann man Systemtreiber benutzen, die nicht Bestandteil der Anwendersoftware, sondern Bestandteil des Betriebssystems sind. Über standardisierte Schnittstellen (Windows GDI, Mac QuickDraw) ruft die Anwendersoftware den Systemtreiber auf und kann damit druckfähige Seiten erzeugen. Diese Art der Erzeugung von PS-Dateien hat den Vorteil, daß alle Anwendungsprogramme auf einen Treiber zugreifen, der sich austauschen, verbessern und an ein bestimmtes Ausgabegerät anpassen läßt.

Die dritte Möglichkeit bieten uns die Anwendungsprogramme selbst, indem sie eigene PS-Treiber zur Verfügung stellen. Das hat zur Folge, daß man vom Betriebssystem unabhängig wird, denn MS-DOS und

```
298 421 translate
% Schritt 1
0 0 144 90 270 arc          % schwarzer Halbkreis
fill
% Schritt 2
0 -72 72 0 360 arc          % unterer schwarzer Kreis
fill
% Schritt 3
gsave                        % zeichne zwei weiße Kreise
1 setgray
0 72 72 0 360 arc           % oberer weißer Kreis
fill
0 -72 12 0 360 arc          % weißer Punkt
fill
grestore
% Schritt 4
0 72 12 0 360 arc           % schwarzer Punkt
fill
0 0 144 0 360 arc           % schwarze Umrandung
stroke
/Times-Roman findfont 20 scalefont setfont
-60 -200 moveto
1 2 scale
(T'AI CHI T'U)show
showpage
```

Abb.1: Skript einer PostScript-Datei [3]

Unix kennen zwar die Bildschirmausgabe, unterstützen aber keine Druckerausgabe. Aber auch unter Windows und Mac OS laufen solche Softwarepakete, die dann den Systemtreiber umgehen (Pass-Through-Modus). Zu diesen Programmpaketen gehören z.B. die Adobe-Produkte *Illustrator*, der intern mit PostScript arbeitet und in der Lage ist, ein viel effektiveres PS-File auszugeben als ein Treiber, und *Photoshop*, der Seiten mit hohen Anforderungen an Farbe und Darstellung erzeugt, die ein Systemtreiber mitunter nicht exakt genug wiedergibt.

Als viertes und letztes gibt es noch PS-Konverter oder auch Filter genannt. Sie können ein bestimmtes Eingangsformat nach PostScript übersetzen. Man unterscheidet hier Text und Grafikkonverter, die Textdateien oder Grafikdateien (z.B. TIFF, GIF, JPEG) nach PostScript übertragen. Vielfach ist das Ergebnis eine EPS-Datei (Encapsulated PostScript).

Als Ergebnis erhält man nun eine geräteunabhängige PS-Datei, die überall gedruckt oder auf andere Geräte ausgegeben werden kann. Es stellen sich aber folgende Fragen: Sind die erzeugten PS-Files wirklich geräteunabhängig? Was passiert, wenn ich das Betriebssystem wechsele oder die Datei nicht sofort drucke? Wie kann ich PS-Dateien mitteilen, um welches Ausgabegerät es sich handelt und wie die damit verbundenen Ausgabevorteile (z.B. Farbräume, Fonts, Druckmedien) zu nutzen sind?

Adobe stellt anderen Entwicklern von PS-Treibern das DSC (document structuring convention) zur Lösung des ersten Teils der Fragen zur Verfügung. Wenn sich nun die Hersteller von Treibern an diese Programmierungskonventionen halten würden, erhielten die Anwender ein geräteunabhängiges PS-File, das sich problemlos ausgeben ließe. Doch ein großer Teil der Entwickler hält sich leider nicht an diese Vereinbarungen. Diese DSC enthalten keine neuen PS-Anweisungen, sondern legen nur fest, welche Struktur PS-Dateien haben müssen. Die wird übersichtlich durch eine Anfangs- und eine Endkommentarzeile gekennzeichnet, erkennbar an der %% Kennzeichnung am Zeilenbeginn. Diese werden dann vom Interpreter ausgewertet.

```

%!PS-Adobe-3.0
...DCS-Kommentare...
%%EndComments

%%Begin Prolog

...
%%EndProlog

%%BeginSetup
...PostScript-Code und Kommentare...
%%EndSetup

%%Page: 1 1

...
%%Page: 2 2

...

%%Trailer
...PostScript-Code und Kommentare

```

Abb.2: Struktur einer PS-Datei unter Anwendung von DSC [1]

Die Möglichkeit, gerätespezifische Funktionen zu nutzen, würde natürlich dem Prinzip der Geräteunabhängigkeit widersprechen. Adobe stellt dazu die PPD-Datei (PostScript Printer Description) zur Verfügung. Diese Datei wird vom Hersteller geliefert und kann als ein kleiner Teil eines PS-Treibers aufgefaßt werden. Sie enthält standardisierte Tabellen und PS-Anweisungen. Der erste Teil eines PS-Treibers liefert also die geräteunabhängige Seite, und der zweite Teil gibt uns die Möglichkeit, geräteabhängige Einstellungen zu definieren. PPD-Dateien sind Textdateien, die verändert werden können.

Wenn man sich für eine dieser Varianten zur Erzeugung von PS-Dateien entscheiden soll, sind wohl, da ja Adobe der PS-Entwickler ist, die aktuellen Adobe-PS-Treiber und die Adobe-Software, die PS-Dateien aus-

gibt, am sichersten, was die Fehleranfälligkeit beim Weiterverarbeiten anlangt.

Spooler

Ein Spooler wird benötigt, um mehreren Nutzern den Zugriff auf ein oder mehrere Drucker zu ermöglichen. Er sammelt die Druckerjobs aus den Anwendungsprogrammen in einer Warteschlange und bereitet sie für die Ausgabe auf. Er fügt z.B. bei unvollständigen PS-Dateien Fonts oder Prologe ein. Wenn einmal ein Druckjob zu einem anderen Drucker umgeleitet werden muß, kann er auch die geräteabhängigen Informationen verändern. Wenn ein Drucker in der Lage ist, Fehlermeldungen zu senden, nimmt ein gut programmierter Spooler diese entgegen und kann darauf reagieren. Der Spooler ist also ein Organisationsprogramm.

Übertragung zum Ausgabegerät

Für den Anschluß von Ausgabegeräten stehen uns verschiedene Schnittstellen zur Verfügung, die sich durch

- Geschwindigkeit
- Bidirektionalität
- serielle oder parallele Übertragung
- lokale oder vernetzte Übertragung

unterscheiden. Die bekanntesten sind die serielle Schnittstelle RS-232, die parallele Schnittstelle von Centronics, AppleTalk, SCSI und Ethernet. Neben der Übertragung von Ausgabedaten werden auch Steuerdaten übertragen. Sie dienen der Synchronisierung der Übertragung, d.h. es wird dafür gesorgt, daß keine Daten verloren gehen, um den Anfang und das Ende eines Jobs zu erkennen und damit den Interpreter wieder in den Ausgangszustand zurückzusetzen, um neue Jobs zu verarbeiten. Manche Drucker können mehrere Sprachen verarbeiten, deshalb muß auch die Sprache erkannt werden. Dazu dienen Übertragungsprotokolle für die jeweiligen Schnittstellen, die hier jedoch nicht weiter betrachtet werden sollen.

Interpreter

Zur Aufbereitung einer PS-Datei für ein Ausgabegerät braucht man einen Interpreter oder auch RIP (Raster Image Processor) genannt. Der Interpreter ist ein Programm, das die PS-Anweisungen ausführt und eine Ausgabe erzeugt. Dabei hängt es vom jeweiligen Ausgabegerät ab, ob der Interpreter Rasterformate (z.B. TIFF, GIF, PICT) oder objektorientierte Grafikformate (z.B. WMF, PICT mit Objekten, CGM) erzeugt. Außerdem muß der Interpreter auch Verwaltungsaufgaben bewältigen: die Kommunikation mit dem Computer, die Verwaltung von Druckerfestplatten und die Steuerung der Druckmechanik. Mit manchen Interpretern kann man auch direkt kommunizieren, d.h. sie können über Tastatur eingegebene oder als Druckjob getarnte PS-Anweisungen ausführen. Damit lassen sich

Druckereinstellungen vornehmen oder PS-Anweisungen testen. Der Interpreter ist also sozusagen das Betriebssystem des Ausgabegeräts.

RIPs oder Interpreter finden wir in verschiedener Konfiguration:

- hardwaremäßig als PS-Controller auf einer Platine im Drucker mit Speicher, Prozessor und ROMs mit dem Interpreter,
- als Hardware-RIP, meist als zusätzlicher Rechner mit einem eigenen nicht üblichen Betriebssystem und dem Interpreterprogramm,
- oder als Software-RIP, denn ein Interpreter braucht, wie aus den anderen beiden Konfigurationen ersichtlich, Prozessor und Speicherplatz. Beides ist ja schon auf einem Rechner verfügbar. Da bietet es sich an, die Interpretation auf den Rechner zu verlegen. Die meisten Software-RIPs können gängige Ausgabegeräte ansteuern. Einer der bekanntesten Software-RIPs ist Ghostscript, welches kostenlos für alle gängigen Betriebssysteme im Internet zur Verfügung gestellt wird. Ghostscript hat den großen Vorteil, daß es PostScript auf dem Bildschirm darstellen kann und uns somit die Möglichkeit gibt, die zu druckende Seite zu begutachten.

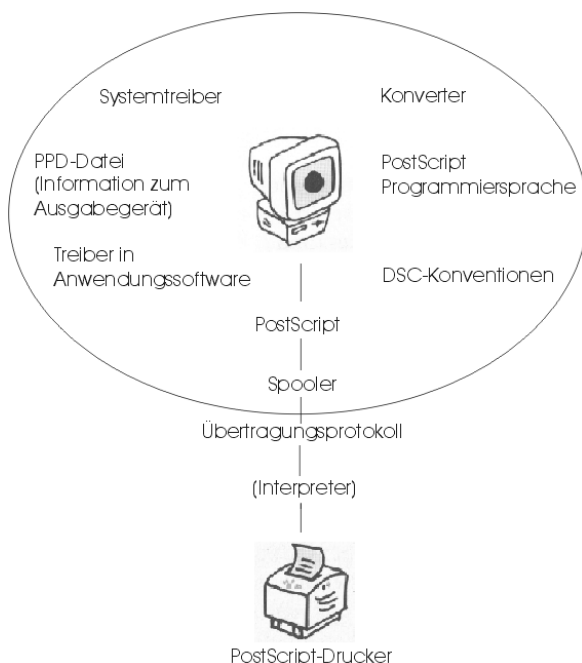


Abb.3: Vom PS-Programm bis zum Drucker
(Die Anordnung des Interpreters ist, wie beschrieben, vom Drucker abhängig)

PostScript-Fehler

In der Praxis treten immer wieder Fehler beim Ausdrucken von PS-Dateien auf, oder man muß erst einige Zeit aufwenden, um eine PS-Datei zu manipulieren,

damit sie sich ausdrucken läßt. Und trotzdem gibt es immer wieder Fälle, bei denen sich eine Datei absolut nicht drucken lassen will. Wenn man sich den beschriebenen Weg von der Erstellung einer PS-Datei bis zum Druck ansieht, warten an vielen Stellen Fehlermöglichkeiten. Fehler treten z.B. auf, wenn man selbst programmiert, wenn Treiber sich nicht an DSC halten oder wenn Anwendungssoftware geräteabhängige Informationen in die Ausgabedateien einbinden. Aber auch Speicherplatzprobleme von Ausgabegeräten – PS-Dateien können sehr groß werden – oder Fehler bei der Datenübertragung können Fehlerursachen sein.

Einer der häufig auftretenden Fehler, der sich auch einfach beheben läßt, ist das Nichterkennen der ersten Zeile einer PS-Datei durch den Interpreter, was zur Folge hat, daß der Quelltext der PS-Datei als ASCII-Text ausgegeben oder die Datei gar nicht erst zum Drucker gesendet wird, weil der Spooler sie nicht als PS-Datei erkennt. Die erste Zeile einer PS-Datei muß immer wie folgt aussehen:

%%!, %!PS, oder %!PS-Adobe-3.0.

Fehlermeldungen erscheinen auf dem Bildschirm oder auf dem Drucker. Fehlermeldungsroutrinen sorgen für eine möglichst verständliche Fehlermeldung, aber wenige lassen sich vom Anwender beheben.

Fehlermeldungen haben die Form:

```
%%[Error: <Fehlernamen>; OffendingCommand:
<Operator> ]%%
```

In der Literatur [1] sind einige häufig auftretende Fehler und Fehlermeldungen und auch Möglichkeiten zu ihrer Beseitigung beschrieben.

EPS

EPS (Encapsulated PostScript) ist einige Jahre später entwickelt worden, um als Standardformat einen Datenaustausch von Grafiken und anderen Objekten zwischen unterschiedlicher Anwendersoftware und unterschiedlichen Betriebssystemen zu ermöglichen. EPS beruht auf PostScript und soll genauso geräteunabhängig sein. Unter diesen Voraussetzungen wird klar, daß das Einsatzgebiet von EPS-Dateien ein anderes ist als das von PS-Dateien. Im Allgemeinen werden fertiggestellte Grafiken und Objekte in EPS-Dateien abgespeichert. Bei der Weiterverarbeitung werden diese Dateien meist in Layouts eingefügt oder ausgedruckt, wobei meistens nur noch eine Größenanpassung erfolgt.

Was nun den Unterschied zwischen PostScript und EPS betrifft, gibt es in der Literatur unterschiedliche Meinungen. Die Abbildung 4 [1] zeigt eine Gegenüberstellung von EPS und PostScript, die bei genauem Hinsehen deutlich macht, daß eine EPS-Datei genauso aussehen kann wie eine PS-Datei.

Probleme mit EPS-Dateien

Da EPS-Dateien durch Software erzeugt werden, ist eine Veränderung des Inhalts meist nicht mehr erforderlich. Aber trotzdem gibt es hin und wieder Probleme mit EPS-Dateien, die es erfordern, eine EPS-Datei zu verändern, um Darstellung und Druck der Datei zu ermöglichen. Im folgenden sollen häufig auftretende Fehler beschrieben und Tips zu Ihrer Beseitigung gegeben werden:

- Wenn eine EPS-Datei erstellt wird, muß das erzeugende Programm die DCS-Konventionen einhalten, d.h., eine EPS-Datei beginnt mit

```
%!PS-Adobe-3.0 EPSF 3.0,
```

versteht Kommentarzeilen, die mit %% beginnen, und endet mit

```
%%Trailer.
```

Eine Fehlerquelle hierbei ist, daß einige Softwarehersteller sich nicht an diese Konventionen halten und vor der ersten und nach der letzten Zeile Informationen speichern. Das hat zur Folge, daß solche Dateien nicht vom Drucker erkannt werden oder nicht von anderen Softwarepaketen weiterverarbeitet werden können. Diese Informationen kann man löschen. Damit ist die Möglichkeit der Weiterverarbeitung gegeben, aber es kann auch passieren, daß das Erzeugerprogramm diese Dateien nicht mehr lesen kann.

- Besonders häufig hat man folgenden Effekt: ein Drucker hat die EPS-Datei erkannt, zeigt an, daß er arbeitet, aber druckt dann nicht. Das liegt meist an der fehlenden, zur EPS-Datei gehörenden Zeile

showpage,

die möglichst am Ende der EPS-Datei, die Struktur beachtend, eingefügt werden muß. Sie dient zur Ausgabe einer Seite auf einen Drucker nach deren Aufbereitung durch den Interpreter.

- Eine Besonderheit von EPS-Dateien ist der BoundingBox-Kommentar

```
%%BoundingBox linksuntenX linksuntenY  
rechtsobenX rechtsobenY
```

Die X- und Y-Werte sind ganzzahlige Koordinaten, die in Punkten angegeben werden. Ein Punkt, so wurde als verbindliche Einheit festgelegt, ist 1/72 Zoll $\approx 0,35$ mm, das bedeutet für eine Seite im A4-Format (210x297mm) 595 x 842 Punkte. Manche Programme geben anstelle von ganzen Zahlen Gleitkommazahlen an, oder es treten negative Zahlen auf, die dann beim Einlesen durch andere Software oder beim Druck zu Fehlern führen können. Die BoundingBox-Koordinaten lassen sich modifizieren.

- Eine Besonderheit von EPS-Dateien ist der Preview-Teil. Verfügt ein verarbeitendes Programm über keinen Interpreter, läßt sich die EPS-Datei nicht am Bildschirm darstellen. Unter Unix ist meist DPS (Display PostScript) implementiert, womit PS-Befehle sofort darstellbar sind. Um die Darstellung ohne Interpreter auch unter anderen Betriebssystemen zu ermöglichen, wird die EPS-Datei durch einen Preview-Absatz ergänzt, den man beim Erstellen von EPS-Dateien durch Menüauswahl hinzufügen kann. In den meisten Fällen handelt es sich um ein Rasterbild, z.B. im TIFF-Format. Die weiterverarbeitende Software kann dann das Rasterbild auf dem Bild-

	EPS	"normale" PostScript-Datei
Anzahl der Seiten	1	beliebig
Ausgabe auf PostScript-Drucker möglich?	ja, eventuell muß der Preview-Teil entfernt werden	ja
Ausgabe auf nicht PostScript-fähigem Drucker möglich?	ja, falls Preview vorhanden	nein
Bildschirmausgabe ohne PS-Interpreter möglich?	ja, falls Preview vorhanden	nein
Datei ist DSC-kompatibel?	ja	empfohlen
Datei enthält BoundingBox-Kommentar?	ja	empfohlen
Kritische Operatoren erlaubt?	nein	ja
Datei enthält showpage?	nicht immer	ja
Austausch zwischen verschiedenen Betriebssystemen möglich?	EPSI oder EPS ohne Preview: ja Mac- oder Windows-EPS mit Preview: nur eingeschränkt	nur drucken
Erzeugung durch Druckertreiber möglich?	mit Einschränkungen	ja

Abb. 4: Gegenüberstellung von EPS und PostScript

schirm darstellen. Kommen nicht PS-fähige Drucker zum Einsatz, wird dieses Rasterbild auch gedruckt (Achtung: Qualitätsverlust durch die geringere Bildschirmauflösung!). Der Preview-Teil kann aber auch Fehlerquelle sein, wenn ein PS-Drucker unverständliche, meist nicht lesbare Fehlermeldungen sendet. Dann sollte man den Preview-Absatz von Anfang bis Ende entfernen, muß sich aber im klaren darüber sein, daß man damit auf die Darstellbarkeit am Bildschirm verzichtet. Es sei noch erwähnt, daß anstelle von Rastergrafiken als Preview auch objektorientierte Darstellungen im Preview (Windows WMF-Dateien; MacOS PICT-Dateien) möglich sind.

- **Ctrl-D Problem**

Das Steuerzeichen Ctrl-D wird am Anfang und Ende einer PS-Datei vom Treiber hinzugefügt, um Anfang und Ende bei der Übertragung zum Drucker über eine parallele oder serielle Schnittstelle zu kennzeichnen. Das funktioniert, solange solche Dateien seriell oder parallel übertragen werden. Wenn man sie aber in eine EPS-Datei einbettet, würden sie den Ausdruck der Datei verhindern. Man erkennt diesen Fehler daran, daß z.B. Ghostscript die Datei am Bildschirm anzeigen kann, sie aber nicht gedruckt wird. Man muß die Ctrl-D-Steuerzeichen aus dem Dateitext entfernen.

- Zum Schluß sei darauf hingewiesen, daß EPS-Dateien nur eine Druckseite enthalten dürfen. Weiterhin kann es Probleme mit allen PS-Operatoren geben, die keine Grafik beschreiben, sondern Hinweise für den Interpreter enthalten (z.B. Papierschacht auswählen, Seitenformat wählen). Das hat zur Folge, daß Seiten nur teilweise ausgedruckt werden.

PDF: PostScript fürs Web oder die neue Druckvorstufe?

Warum PDF?

„Adobe Acrobat und das Portable Document Format (PDF) versprechen den universellen Austausch von Dokumenten, unabhängig von Betriebssystem und Anwendungsprogramm. Dank Hypertext-Funktionen sind die digitalen Dokumente ihren konventionellen Vorgängern sogar überlegen.“ (aus [1], Seite 268.)

So oder ähnlich werben Artikel und Bücher für PDF als universelles Austauschformat für Dokumente. Was ist nun tatsächlich dran am vielgelobten PDF?

Das Portable Document Format (PDF) wurde von der Firma Adobe konzipiert und ist, anders als PostScript, ein Dokumentenformat und keine Programmiersprache. Während PostScript für die Druckvorstufe, d.h. die Visualisierung von Dokumenten für das Drucken, konzipiert wurde, soll PDF dem Austausch von Dokumenten über Hardware- und somit Betriebssystemgrenzen dienen und unabhängig von den verwendeten Programmen sein. PDF-Dokumente lassen sich sowohl

unter MacOS als auch auf den Microsoft Windows-Plattformen und unter UNIX bzw. LINUX bearbeiten. Mit der neuen Version 1.3 bringt Adobe PDF nun auch als neuen Standard für die Druckvorstufe ins Spiel.

Als Szenario wird häufig auf die Metapher des digitalen Büros zurückgegriffen. In dieser wird zum Beispiel ein Werbeblatt für ein Produkt entworfen, dieses dann als PDF-Dokument per E-Mail an Kollegen zur Kontrolle verschickt und von letzteren dann mit Notizen versehen, die direkt an das Dokument angebracht werden. Nach einer weiteren Bearbeitung kann der fertig gestaltete Flyer nun vom Grafiker zum Setzer und so zur Druckerei transportiert werden, ohne daß Probleme mit fehlenden Schriftarten, unterschiedlich benutzten DTP-Programmen, etc. auftreten. Gleichzeitig kann dasselbe PDF-Dokument direkt ins WWW gestellt werden, da es bereits fertig gestaltet über spezielle Plugins, wie dem Adobe Acrobat Reader, lesbar ist.

Mehr als PostScript?

Die folgenden Probleme bei der Verwendung von PostScript erschweren es, dieses Format als Austausch- oder Archivierungsformat zu benutzen. Zum einen können PS-Dateien sehr groß werden. Ihre Bearbeitung erfordert zumeist sehr umfangreiche Programme. Dadurch ist es schwer, einzelne Elemente auf einer Seite zu verändern oder zu manipulieren (extrahieren, hinzufügen). Auch die Suche nach Textstellen in PS-Dateien ist beschränkt. Ein weiterer Nachteil ist auch, daß bei der Verwendung von PostScript spezielle Schriften notwendig sind, ohne die die Dokumente unbrauchbar werden. Um dem Abhilfe zu schaffen, faßte die Firma Adobe daher zahlreiche Entwicklungen zusammen und definierte 1993 das Dokumentenformat PDF, siehe [6].

Das Dateiformat PDF

Das aktuell in der Version 1.3 vorliegende Format entspricht von der Abbildungsleistung her PostScript 3, vgl. [4]. Adobe nutzt hier viele Bestandteile aus anderen Entwicklungen (siehe [1], Seite 271), wie etwa die Grafikfähigkeit von PostScript, die Kompaktheit des Dateiformates von Adobe Illustrator, die Kompressionsverfahren und die gerätenabhängige Farbe aus PostScript Level 2 oder auch Type-1- und Multiple-Master-Fonts sowie das in PostScript 3 integrierte CIO-Format für Double Byte Schriften, wodurch die Darstellungsmöglichkeiten für asiatische Zeichen verbessert werden. Dazu kommen Hypertext-Methoden, wie z.B. Sprungverweise oder Strukturierung von Dokumenten, ähnlich wie in SGML oder XML.

Eine PDF-Datei besteht aus mehreren Teilen, vgl. [5]:

- einem Vorspann
- dem Körper der PDF-Datei
- der Referenztabelle (Cross Reference Tabelle)
- dem Nachspann

Dabei enthält eine PDF-Datei immer nur einen Vorspann, die Folge Körper, Cross Reference Table, Nachspann kann jedoch mehrfach auftreten. Wird eine PDF-Datei bearbeitet, werden die modifizierten Dokumentteile (Body, Cross Reference und Nachspann) immer hinter die letzte %%EOF -Zeile geschrieben.

Der Vorspann eines PDF-Dokumentes besteht aus folgenden Zeilen:

```
%PDF-1.2
%âãĀŒ
```

wobei PDF-1.2 die benutzte PDF-Version angibt.

Der Körper des Dokumentes besteht dabei aus einer Folge von Objekten. Diese enthalten entweder Inhalte oder sie beschreiben Elemente, wie z.B. Fontdefinitionen. Es gibt direkte Objekte, wie numerische Konstanten, Strings oder Namen, z.B.:

```
/Length 1 8
```

oder indirekte, z.B.:

```
/Length 7 0 R
```

```
7 0 obj
9628
endobj
```

Indirekte Objekte sind meist komplexer, sie enthalten oft binär codierte Streams, z.B.:

```
6 0 obj
<<
/Length 7 0 R
/Filter /FlateDecode
>>
stream
H%o"W9'$7|Á_al __IrTĪ_ß
.
.
.
endstream
endobj
```

sie beginnen mit *obj*, enden mit *endobj* und werden durch den Operator R referenziert.

PDF ist also nicht rein ASCII-orientiert, da es über Kompressionsverfahren verfügt, die auch Nicht-ASCII-Code erzeugen können und binär kodierte Streams, z.B. JPEG-Streams, erlaubt sind.

Die Cross Reference gibt an, welche Objekte mit welchen Offsets im PDF-Strom enthalten sind. Sie besitzt eine definierte Datenstruktur. Jede Zeile muß genau zwanzig Zeichen lang sein (inkl. Zeilenende), der Buchstabe am Ende einer Zeile gibt dabei an, ob das Objekt frei 'f' oder belegt 'n' ist. Freie Objekte entstehen z.B. durch Löschen.

```
xref
0 10
0000000000 65535 f
0000009981 00000 n
0000010255 00000 n
```

Der Offset der Cross Reference Table wird wie folgt angegeben:

```
startxref
10614
```

Im Nachspann (Trailer) werden Metainformationen angegeben. Möglich sind:

```
/Keywords
/Subject
/Title
/Author
/ModDate
/CreationDate
/Producer
```

Spezielle Eigenschaften von PDF

Grafikmöglichkeiten: PDF arbeitet mit dem Grafikmodell von PostScript. Aufgrund der Tatsache, daß PDF keine Programmiersprache ist, ist seine Flexibilität, die graphische Darstellung betreffend, jedoch eingeschränkt. Es gibt keine Kontrollstrukturen, wie z.B. Schleifen oder Abfragen. Dadurch ist PDF aber auch effizienter als PostScript, die Dateien sind kleiner und lassen sich einfacher auswerten. Die PDF Version 1.3 kann den vollen Sprachumfang von PostScript 3 speichern. Neu sind auch einige technische Parameter, so kann die Höhe und Breite einer PDF-Seite nun zwischen 1 Millimeter und 500 Zentimeter betragen, Vektorgraphiken werden in einer Auflösung bis zu 4000 dpi unterstützt. Neue Technologien für das Farbmanagement wurden ebenso integriert, vgl. [7], wie die Einbettung von Schriften.

Hypertext-Funktionalitäten: PDF bietet die Möglichkeit, *Lesezeichen* (bookmarks) zu benutzen, die hierarchisch geordnet sein können. Diese werden dann in Acrobat in einem extra Fenster dargestellt und können als Inhaltsverzeichnis fungieren. Beim Klicken auf ein Element dieses Inhaltsverzeichnisses wird dann zur entsprechenden Überschrift bzw. referenzierten Seite gesprungen. Ähnlich können Thumbnails, *Vorschau- grafiken einzelner Seiten* erstellt werden. *Querverweise* innerhalb eines Dokumentes können ebenso in PDF-Dokumente integriert werden, wie Hyperlinks zu Dokumenten im WWW. Eine besondere Möglichkeit ist die, *Anmerkungen* an Dokumenten anzubringen, um Dokumente um eigene Notizen zu ergänzen und so die Kommunikation zu anderen Personen zu vereinfachen. Annotationen sind Notizen oder andere Objekte, die mit der Seite assoziiert werden, jedoch unabhängig von der Seitenbeschreibung an sich sind. PDF unterstützt unterschiedliche Arten von Annotationen: Textnotizen,

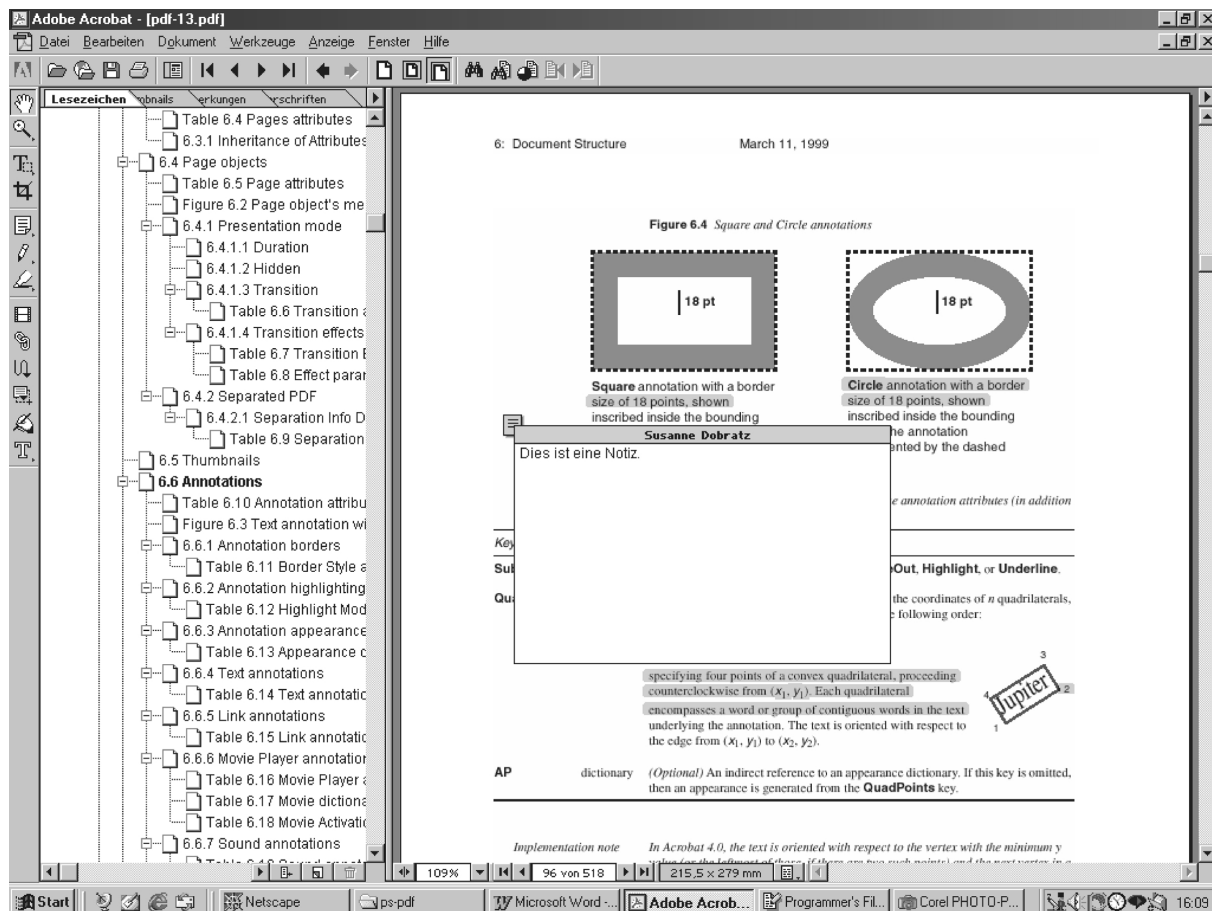


Abb. 6: Beispiel für die Nutzung unterschiedlicher Notiz-funktionen

Hypertext Links, Bewegtbilder (Movies), Audioanmerkungen etc. Weitere Annotationsmöglichkeiten, wie Stempel, Linien, Rechtecke, Kreise, Textdurchstreichungen, Highlighting von Elementen und Unterstreichungen, sind möglich.

Die Erstellung von **Formularen** in PDF ist mit einem speziell dafür definierten Format FDF (Forms Data Format) möglich. Es dient der Übertragung von Formulardaten auf einen Server und deren Zusammenbau zu einem Formular. Das FDF basiert auf dem PDF-Format und soll über Webbrowser mit der Angabe des MIME-Types `application/vnd.fdf` aufgerufen werden können. Ebenso gibt es jetzt eine integrierte Javascript-Unterstützung.

Sicherheitsfunktionen: Die neueste PDF-Version läßt die Benutzung digitaler Signaturen zu. Eine vorhandene digitale Signatur wird in PDF dargestellt als stream. Dieser enthält die ungeprüfte (*invalidated*) digitale Signatur in Form von Text, Bitmaps eines elektronischen Fingerabdruckes oder die spezielle Repräsentation der digitalen Signatur, die durch bestimmte Signaturverfahren festgelegt ist. Bei der Anzeige oder beim Druck wird dieses Feld wie alle anderen angezeigt. Damit ist die Anzeige der Signatur unabhängig von den vorhandenen Plugins des PDF-Browsers. Der

Browser soll diese Signatur dann überprüfen und sie entsprechend kennzeichnen als gültig (*valid*) oder ungültig (*invalid*).

Strukturierung von PDF-Dokumenten: Ein weiteres neues Merkmal von PDF 1.3 ist die Möglichkeit, Strukturinformationen, wie die Organisation von Kapiteln und Unterkapiteln, zu repräsentieren. Diese logische Struktur eines Dokuments ist ähnlich der standardisierter Markup Sprachen, wie HTML, SGML oder XML aufgebaut. Diese logischen Strukturelemente können sowohl die eigentlichen Inhalte als auch Attribute beinhalten. Dabei wird die logische Struktur eines PDF-Dokuments unabhängig von seinem sichtbaren Inhalt (den graphischen Objekten) gespeichert. Es gibt nur Zeiger, die von der logischen Struktur auf die physikalische Repräsentation verweisen und umgekehrt.

Werkzeuge für PDF

Viele der beschriebenen Merkmale, die bereits schon in früheren PDF-Definitionen vorgesehen waren, werden erst durch die Implementation von Werkzeugen für ihre Benutzung zum Leben erweckt. Für PDF ist dies vor allem das Produkt Adobe Acrobat mit seinen verschiedenen Zusatzmodulen, welches die Nutzung von PDF erst richtig möglich macht.

Das Erstellen von PDF-Dokumenten geschieht entweder direkt aus einem von Adobe unterstütztem An-

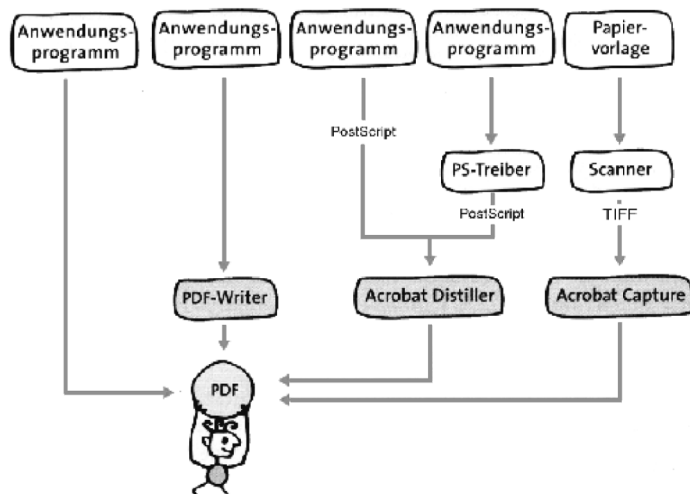


Abb. 7: Möglichkeiten der PDF-Erstellungen

wendungsprogramm heraus mit dem PDF Writer, der für die Microsoft Office Produkte Word, Excel, PowerPoint zur Verfügung steht, oder mittels Acrobat Destiller über den Umweg PostScript. Letzterer ist, obwohl umständlicher, der derzeit sicherste Weg, sauberes PDF zu erzeugen. Bei Word hat die Benutzung des PDF Writer zum Beispiel den Vorteil, daß bei Verwendung von Formatvorlagen Elemente wie Überschriften automatisch in Lesezeichen und Anmerkungen in PDF-Notizen umgewandelt werden. Gängige Adobe Produkte, wie Photoshop, PageMaker oder Illustrator, können ebenfalls in PDF speichern. Empfehlenswert ist jedoch nur der Weg über den Destiller.

Ein besonders schönes Werkzeug, welches mit Acrobat 4 mitgeliefert wird, ist das Webcapture-Modul, welches momentan nur für die Microsoft Windows-Version verfügbar ist. Damit lassen sich einzelne Webseiten oder der Inhalt kompletter Webserver in Acrobat öffnen und in PDF abspeichern. Dabei konvertiert das Programm alle HTML-Links in PDF-Links

und überträgt auch Formularfelder, Frames, Tabellen sowie GIF- und JPEG-Bilder.

Weitere Möglichkeiten von Acrobat 4 liegen darin, Dokumente miteinander vergleichen zu können und Veränderungen innerhalb eines Dokuments durch die Rollback-Funktion abrufen zu können, sowie in der Bearbeitbarkeit von Text in PDF-Dateien. Bei Grafiken wird das TouchUp-Tools aktiv, welches Anwendungsprogramme wie Photoshop 5.0 oder Illustrator 8.0 aufruft, in der die Veränderungen der Grafik dann vorgenommen werden können und im aktiven PDF-Dokument automatisch aktualisiert werden.

Adobe hat mit der Definition der Version 1.3 des PDF-Formates den Schritt unternommen, nun auch einen Einsatz des Formates im Prepress und Preprint-Bereich zu etablieren. Bis PDF in dieser Hinsicht jedoch Einzug hält und ob es tatsächlich einen Fortschritt in Richtung Geräteunabhängigkeit darstellt, wird sich zeigen.

Schlußwort

Wir haben versucht, einige Anregungen zur Arbeit mit PostScript und EPS zu geben, sind uns aber bewußt, daß noch wichtige Informationen fehlen, z.B. Angaben zu den verschiedenen PS-Versionen, zu Schriften und Farbdarstellungen. Diese wichtigen Kapitel sollen später nachgeholt werden. Auch gibt es sicher wesentlich mehr zur Anwendung von PDF im Prepress-Bereich und zu den neuen Eigenschaften im Grafikbereich zu sagen. Unterschlagen haben wir hier ebenfalls die Beschreibung des Weges von LaTeX unter Unix/Linux zu sauberem PDF. Auch diese Berichte werden in späteren Artikeln nachgeholt.

Literaturverzeichnis:

- [1] THOMAS MERZ: *Die PostScript-@Acrobat-Bibel*. München: Thomas Merz Verlag, 1996.
- [2] Adobe Systems Inc.: *PostScript Handbuch*. Addison-Wesley (Deutschland) 1989.
- [3] ROSS SMITH: *PostScript*. München: te-wi Verlag GmbH, 1990.
- [4] OLAF DRÜMMER & MICHAEL KARBE: Neu destilliert - PDF 1.3 und Acrobat 4 im Test, in: *MACUP 04/99*, Seiten 196 - 204
- [5] EDGAR HUCKERT: Dokumentencontainer – Das Dokumentenaustauschformat PDF für Adobes Acrobat, in: *c't 1996*, Heft 5, Seiten 316 - 324
- [6] Adobe Systems Incorporated: Portable Document Reference Format, (zuletzt gesehen: 05.07.1999, 10:35), <http://partners.adobe.com/asn/developer/acrosdk/DOCS/pdfspec.pdf>
- [7] OLIVER SCHRÖDER: Erster Blick auf Acrobat 4, in: *Macwelt 4/99*, Seiten 96-100

Susanne Dobratz
susanne.dobratz@rz.hu-berlin.de

Hannelore Schmidt
hschmidt@rz.hu-berlin.de